IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:

Applicants	:	Reuven Bakalash et al.
U.S. Application No.	:	10/579,682
Filing Date	:	November 19, 2004
Title of Invention	:	PC-BASED COMPUTING SYSTEM EMPLOYING PARALLELIZED GRAPHICS PROCESSING UNITS (GPUS) INTERFACED WITH THE CENTRAL PROCESSING UNIT (CPU) USING A PC BUS AND A HARDWARE GRAPHICS HUB HAVING A ROUTER
Group Art No.	:	2628
Examiner	:	Hau H. Nguyen
Attorney Docket No.	:	142-002USAC00

Honorable Commissioner of Patents
    and Trademarks
Washington, DC 20231

SECOND SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT UNDER 37 C.F.R. 1.97

Sir:

    In order to fulfill Applicants' continuing obligation of candor and good faith as set forth in 37 C.F.R. 1.56, Applicants submit herewith a Second Supplemental Information Disclosure Statement prepared in accordance with 37 C.F.R Sections 1.97, 1.98 and 1.99.

    The disclosures enclosed herewith are as follows:

U.S. PUBLICATIONS

NUMBER	FILING DATE	TITLE
7,598,958 B1	November 17, 2004	MULTI-CHIP GRAPHICS PROCESSING UNIT APPARATUS, SYSTEM AND METHOD
7,525,547 B1	December 17, 2004	PROGRAMMING MULTIPLE CHIPS FROM A COMMAND BUFFER TO PROCESS MULTIPLE IMAGES
7,388,581 B1	August 28, 2003	ASYNCHRONOUS CONDITIONAL GRAPHICS RENDERING
6,956,579 B1	August 18, 2003	PRIVATE ADDRESSING IN A MULTI-

PROCESSOR GRAPHICS PROCESSING
SYSTEM

6,362,825 B1   January 19, 1999   REAL-TIME COMBINATION OF
ADJACENT IDENTICAL PROMITIVE
DATA SETS IN A GRAPHICS CALL
SEQUENCE

TECHNICAL PUBLICATIONS

Antonio Garcia and Han-Wei Shen "An Interleaved Parallel Volume Render With PC-clusters",
Proceedings of the Fourth Eurographics Workshop on Parallel Graphics and Visualization, 19
pages, 2002.

Erik Reinhard and Chuck Hansen, "A Comparison of Parallel Compositing Techniques on
Shared Memory Architectures", Accepted for the Eurographics Workshop on Parallel Graphics
and Visualisation, Girona, Spain, September 2000.

Steven Molnar, "Combining Z-buffer Engines for Higher-Speed Rendering", Proceedings of the
1988 Eurographics Workshop on Graphics Hardware, 11 pages, 1988.

John Eyles et al., "PixelFlow: The Realization," SIGGRAPH/EUROGRAPHICS Conference on
Graphics Hardware, 13 pages, 1997.

ABSTRACTS OF DISCLOSURE

U.S. Patent No. 7,598,958 B1 to Kelleher discloses a multi-chip graphics system which
includes a master chip and a slave chip coupled by an interlink. The slave chip performs a
graphics processing operation in parallel with the master chip, improving the performance of the
master chip. In one embodiment, an individual graphics processing unit (GPU) chip includes a
normal operational mode, a master mode, and a slave mode to permit an individual GPU chip to
be used as individual processor or to be packaged as part of a master/slave pair.

U.S. Patent No. 7,525,547 B1 to Diard discloses methods, apparatuses, and systems for
operating a plurality of graphics devices involving using the graphics devices to processes a
sequence of images, wherein at least one first graphics device processes a first image, and at least
one second graphics device processes a second image, communicating a first command
associated with the first image to the at least one first graphics device and the at least one second
graphics device, wherein the first command is to be executed by the at least one first graphics
device and the at least one second graphics device, and communicating a second command
associated with the first image to the at least one first graphics device and the at least one second
graphics device, wherein the second command is to be executed by the at least one first graphics
device but not by the at least one second graphics device.

U.S. Patent No. 7,388,581 B1 to Diard et al. discloses a graphics processing unit which implements conditional rendering by putting itself in a state in which it does not execute any rendering commands. Once the graphics processing unit is placed in this state, all subsequent rendering commands are ignored until another rendering command explicitly removes the graphics processing unit from this state. Conditional rendering commands enable the graphics processing unit to place itself in and out of this state based upon the value of a flag in memory. Conditional rendering commands can include conditions that must be satisfied by the flag value in order to change the state of the graphics processing unit. The value of the flag can be set by the graphics processing unit itself, a second graphics processing unit, a graphics coprocessor, or the central processing unit. This enables a wide variety of conditional rendering methods to be implemented.

U.S. Patent No. 6,956,579 B1 to Diard et al. discloses systems and methods for private addressing in a multi-processor graphics processing subsystem having a number of memories and a number of graphics processors. Each of the memories includes a number of addressable storage locations, and storage locations in different memories may share a common global address. Storage locations are uniquely identifiable by private addresses internal to the graphics processing subsystem. One of the graphics processors is able to access a location in a particular memory by referencing its private address.

U.S. Patent No. 6,362,825 B1 to Johnson discloses a graphics call sequence optimizer for use in a graphics system that includes a display list memory to store graphics calls to be executed. The optimizer optimizes an original graphics call sequence that includes a plurality of graphics primitive data sets generated by a graphics application program in accordance with a graphics application program interface, such as OpenGL, generating an optimized graphics call sequence to be stored in the display list memory. The optimizer is configured to optimize the original graphics call sequence to produce the optimized graphics call sequence without storing the original graphics call sequence in the display list memory. In one embodiment, the optimizer is configured to coalesce graphics primitive data sets within the original graphics call sequence to generate a corresponding single graphics primitive data set in the optimized graphics call sequence that causes a same rendering in the graphics system as the original graphics call sequence. In another embodiment, the optimizer coalesces a series of graphics primitive data sets of a particular primitive type occurring sequentially within the original graphics call sequence into a single graphics primitive data set of the particular type in the optimized graphics call sequence. In a further embodiment, the graphics call sequence optimizer is configured to coalesce graphics primitive data sets of the particular type into a single graphics primitive data set of the particular type until a graphics primitive data set that is not of the particular type appears in the original graphics call sequence. In embodiments where the API is the OpenGL API, each of the plurality of graphics primitive data sets in the original graphics call sequence comprises a glBegin() graphics call, a glEnd() graphics call, and at least one graphics vertex call between the glBegin() graphics call and the glEnd() graphics call. In such an embodiment, the optimizer is configured to remove all glBegin() and glEnd() graphics calls from the original graphics call sequence other than a glBegin() graphics call occurring in the first graphics primitive data set in the original graphics call sequence and a glEnd() graphics call occurring in the last graphics primitive data set in the original graphics call sequence. Method embodiments are also disclosed.

The SIGGRAPH/EUROGRAPHICS Conference presentation (October 17, 2002) entitled "An Interleaved Parallel Volume Renderer With PC-clusters," describes parallel rendering as having been realized using various load distribution methods that subdivide either the screen, called image-soace partitioning, or the volume dataset, called object-space partitioning. The major advantages of image-space partitioning are load balancing and low communication overhead, but processors require access to the full volume in order to render the volume with arbitrary views without frequent data redistributions. Subdividing the volume, on the other hand, provides storage scalability as more processors are added, but requires image compositing and thus higher communication bandwidth for producing the final image. In this paper, we present a parallel volume rendering algorithm that combines the benefits of both image-space and object-space partitioning schemes based on the idea of pixel and volume interleaving. We first subdivide the processors into groups. Each group is responsible for rendering a portion of the volume. Inside of a group, every member interleaves the data samples of the volume and the pixels of the screen. Interleaving the data provides storage scalability and interleaving the pixels reduces communication overhead. Our hybrid object- and image-space partitioning scheme was able to reduce the image compositing cost, incur in low communication overhead and balance rendering workload at the expense of image quality. Experiments on a PC-cluster demonstrate encouraging results.

Publication entitled "A Comparison of Parallel Compositing Techniques on Shared Memory Architectures," describes parallel compositing techniques that have traditionally focused on distributed memory architectures with communication of pixel values usually being the main bottleneck. On shared memory architectures, communication is handled through memory accesses, obviating the need for explicit communication steps. Shared memory architectures with multiple graphics accelerators provide the capability for parallel rendering while combining the partial results from multiple graphics adaptors requires compositing. For this reason, in this paper shared memory architectures are considered for compositing operations. A number of previously introduced parallel compositing algorithms are compared on a shared memory machine, including the binary swap and parallel pipeline techniques.

Publication entitled "Combining Z-buffer Engines for Higher-Speed Rendering," describes a hardware architecture for combining the outputs of a number of z-buffer rendering engines to achieve higher performance than is possible with a single renderer. It allows a combination of renderers to achieve the same price/performance ratio as the individual renderers that compose it, and can be extended to create systems with arbitrarily high performance. The described architecture is based on a fusion of scan-line rendering and the conventional z-buffer algorithm. The frame buffers of several z-buffer engines are modified to scan out z-values as well as color values. Multiplexing devices combine the z/color streams from each pair of frame-buffers. These z/color streams are then combined by further multiplexers, creating a binary tree that funnels the z/color information from the many conventional frame buffers into a single z/color stream. The color stream is then used to drive a standard display device. The proposed architecture allows rendering rates of millions and even tens of millions of polygons per second. The basic architecture can be extended with additional hardware to perform antialiasing and texture-mapping.

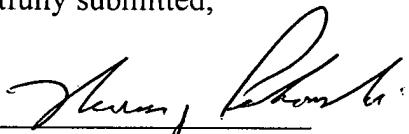Publication entitled "PixelFlow: The Realization," describes an architecture for high-

speed, highly realistic image generation, based on the techniques of object-parallelism and image composition. Its initial architecture was described in [MOLN92]. After development by the original team of researchers at the University of North Carolina, and co-development with industry partners, Division Ltd. and Hewlett-Packard, PixelFlow now is a much more capable system than initially conceived and its hardware and software systems have evolved considerably. This paper describes the final realization of PixelFlow, along with hardware and software enhancements heretofore unpublished.

A separate listing of the above references on PTO Form 1449 as well as hard copies of all non-U.S. references have been enclosed herewith for the convenience of the Examiner.

Enclosed in payment of the requisite fee of $180.00 is Thomas J. Perkowski, Esq., P.C. Check No. 8617. The Commissioner is hereby also authorized to charge any fee deficiencies or overpayments to Deposit Account No. 16-1340.

Respectfully submitted,
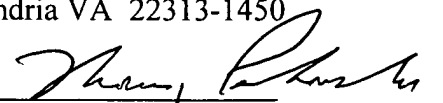
Dated: March 4, 2010

_____
Thomas J. Perkowski
Reg. No. 33,134
Attorney for Applicants
Thomas J. Perkowski, Esq., P.C.
22 Thorndal Circle
Darien, Connecticut 06820
203-357-1950
http://www.tjpatlaw.com

<u>CERTIFICATE OF MAILING UNDER</u>
<u>37 C.F.R. 1.08</u>

I hereby certify that this correspondence
is being deposited with
the United States Postal Service
on March 4, 2010
in a Postage Prepaid envelope as
First Class Mail, addressed to:

Commissioner of Patents
P.O. Box 1450
Alexandria VA 22313-1450

_____
Thomas J. Perkowski, Esq.
Reg. 33,134
Date: March 4, 2010